

FIG. 1

FIG. 2 (Prior Art)

```

Procedure HandleJoin( $gp, n, s$ )
parameters
     $gp$       Multicast group to join
     $n$        Neighbor  $n$  transmitter of request
     $s$        Node originating join request
begin
    if ( $s = i$ )
        return; [ we don't want join loops, so ignore... ]
    endif
     $g \leftarrow \{group\ x \mid x \in MRT_i, x.group = gp\}$ ;
    if ( $g = \emptyset$ )
        [ Group is unknown ]
         $g.group \leftarrow gp$ ;
         $g.status \leftarrow g.status \wedge NOT\_MEMBER$ ;
         $MRT_i \leftarrow MRT_i \cup \{g\}$ ;
    endif
    if ( $i \in CORES_{gp}$ )
        [ this node is one of the cores ]
         $g.status \leftarrow g.status \wedge CORE$ ;
         $core \leftarrow i$ ;
    else
         $core \leftarrow \{node\ k \mid k \in CAM_i^{gp}\}$ ;
    endif
    if ( $core \neq \emptyset$ )
        if ( $isDuplex(i, g)$ )
            call HandleJoinAmDuplex( $gp, n, s$ );
        else
            call HandleJoinAmNotDuplex( $gp, n, core, s$ );
        endif
    endif
end

```

FIG. 3

```

Procedure HandleJoinAmNotDuplex( $g, n, k, s$ )
parameters
     $g$       Multicast group to join
     $n$       Neighbor  $n$  transmitter of request
     $k$       Chosen core for multicast group  $g$ 
     $s$       Node originating join request
begin
    if (  $\exists nb \mid nb \in N_i^g, nb.status = DUPLEX \text{ and } nb \neq n$  )
        [ Any neighbor already a duplex member? ]
         $g.status \leftarrow g.status \wedge DUPLEX$ ;  $g.modified \leftarrow TRUE$ ;
        call HandleJoinAmDuplex( $g, n, s$ );
        return;
    endif
    if (  $PEND_i^g = \emptyset$  )
        [ no pending duplex/simplex join ]
         $nb \leftarrow \text{call NextHop2Core}(k)$ ;
        if (  $nb \neq \emptyset$  )
             $p.address \leftarrow n$ ;
             $p.status \leftarrow p.status \wedge DUPLEX$ ;
             $PEND_i^g \leftarrow PEND_i^g \cup \{p\}$ ;
            if (  $n = i \text{ and } n \notin LR_i^g$  )
                 $lr.address \leftarrow n$ ;
                 $lr.status \leftarrow lr.status \wedge PENDING$ ;
                 $LR_i^g \leftarrow LR_i^g \cup \{lr\}$ ;
            endif
            call send( $JOIN, g, nb, s$ );
        endif
    else [ There is a pending request. ]
         $p \leftarrow \{x \mid x \in PEND_i^g\}$ ;
        if (  $p.address = i \text{ and } n \neq i$  )
             $p.address \leftarrow n$ ; [ Previous request was local ]
        endif
         $p.status \leftarrow p.status \wedge DUPLEX$ ;
    endif
end

```

FIG. 4

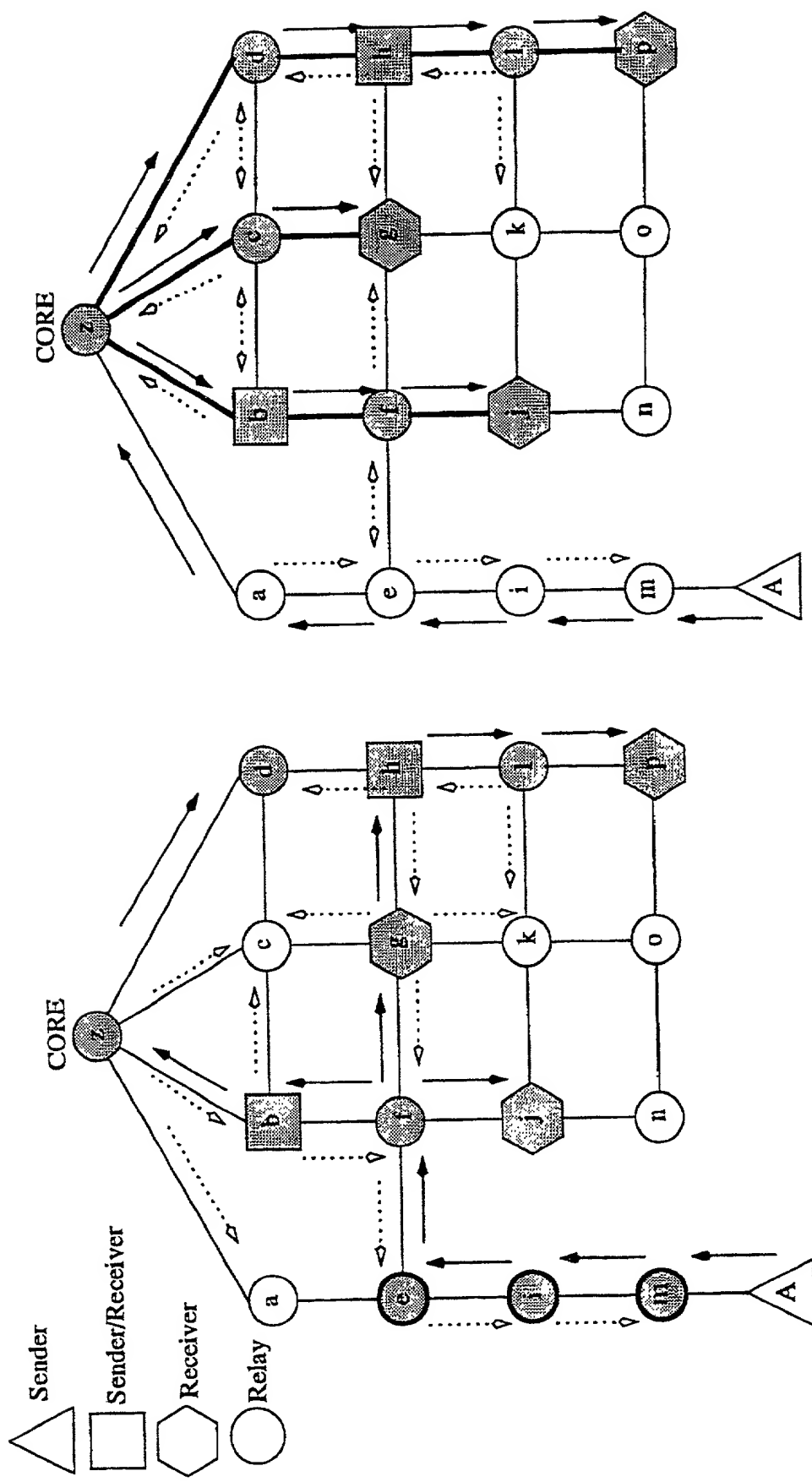


FIG. 6 (Prior Art)

FIG. 5

Procedure HandlePushJoin(*gp, n, s, src*)

parameters

gp Multicast group to join
n Neighbor *n* transmitter of request
s Node originating push join request
src Node that is source of multicast data traffic

begin

if (*s* = *i*)

 return; [no loops, so ignore...]

endif

$g \leftarrow \{group\ x \mid x \in MRT_i, x.group = gp\};$

if ($g = \emptyset$)

 [Group is unknown]

$g.group \leftarrow gp;$

$g.status \leftarrow g.status \wedge NOT_MEMBER;$

$MRT_i \leftarrow MRT_i \cup \{g\};$

endif

if (isDirectlyConnected(*i, src*))

 [Source of traffic is attached to me]

 call HandlePushJoinDC(*gp, n, s, src*);

else

 call HandlePushJoinNonDC(*gp, n, s, src*);

endif

end

FIG. 7

Procedure HandlePushJoinNonDC(g, n, s, src)

parameters

g Multicast group to join
 n Neighbor n transmitter of request
 s Node originating join request
 src Node that is source of multicast data traffic

begin

if ($\{ \exists p \mid p \in PENDPJ_i^g, p.sender = src \}$)
 [Ignore PJ for an existing sender,]
 [but update info if this node started pj]
 if ($p.address = i$)
 $p.address \leftarrow n$;
 endif

else

$nb \leftarrow \text{call NextHop}(src)$;
 if ($nb \neq \emptyset$)
 $p.address \leftarrow n$;
 $p.sender \leftarrow src$;
 $p.status \leftarrow p.status \wedge NOT_MEMBER$;
 $p.anchor \leftarrow NOT_ANCHOR$;
 $PENDPJ_i^g \leftarrow PENDPJ_i^g \cup \{p\}$;
 call send($PUSH_JOIN, g, nb, src, s$);
 endif

endif

endif

end

FIG. 8

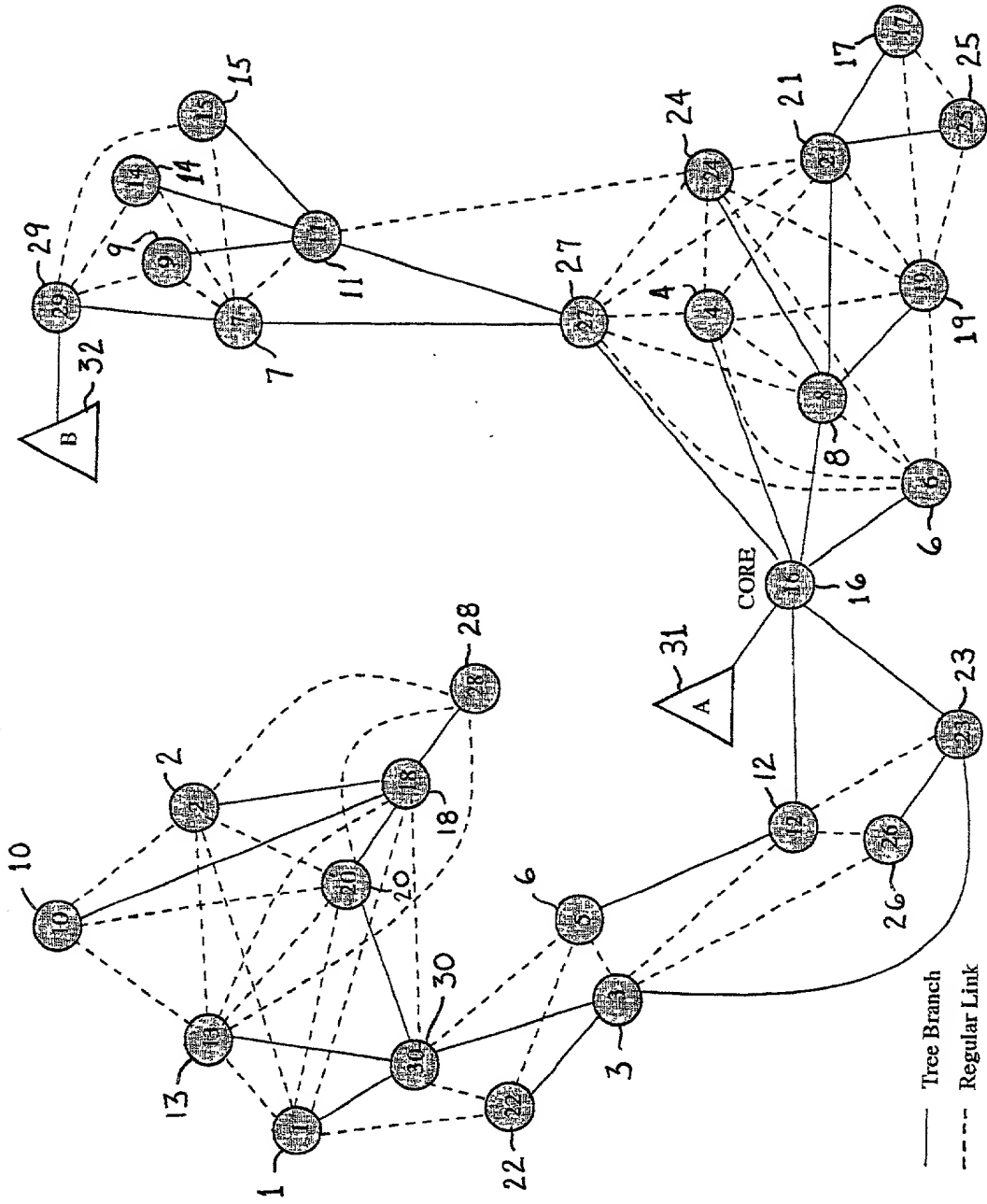


FIG. 9

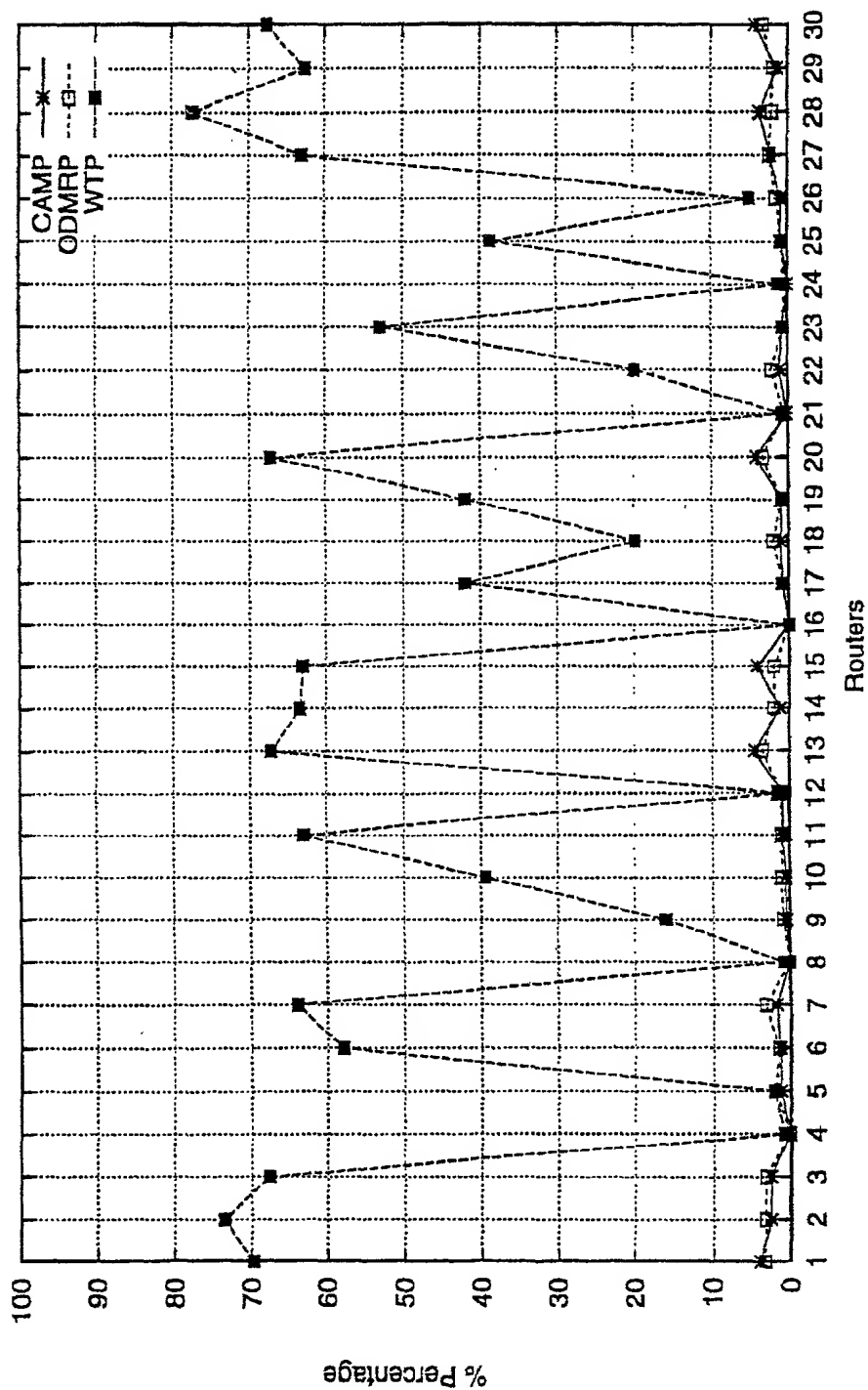


FIG. 10

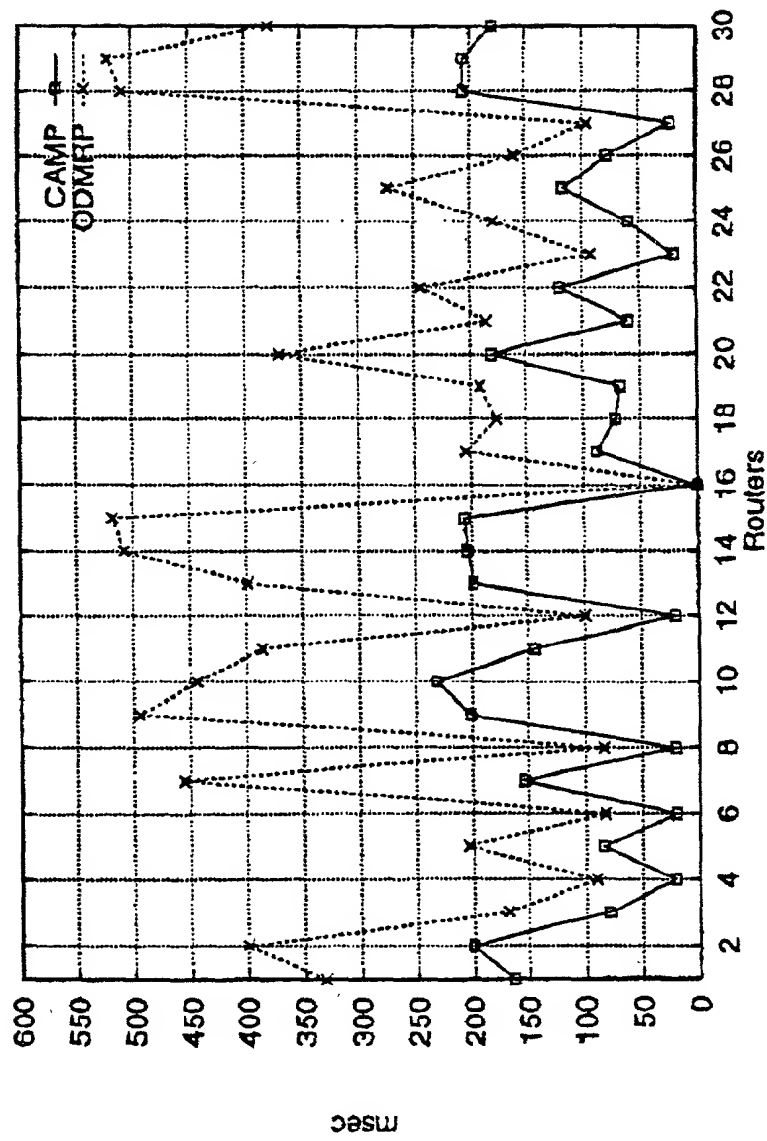


FIG. 11

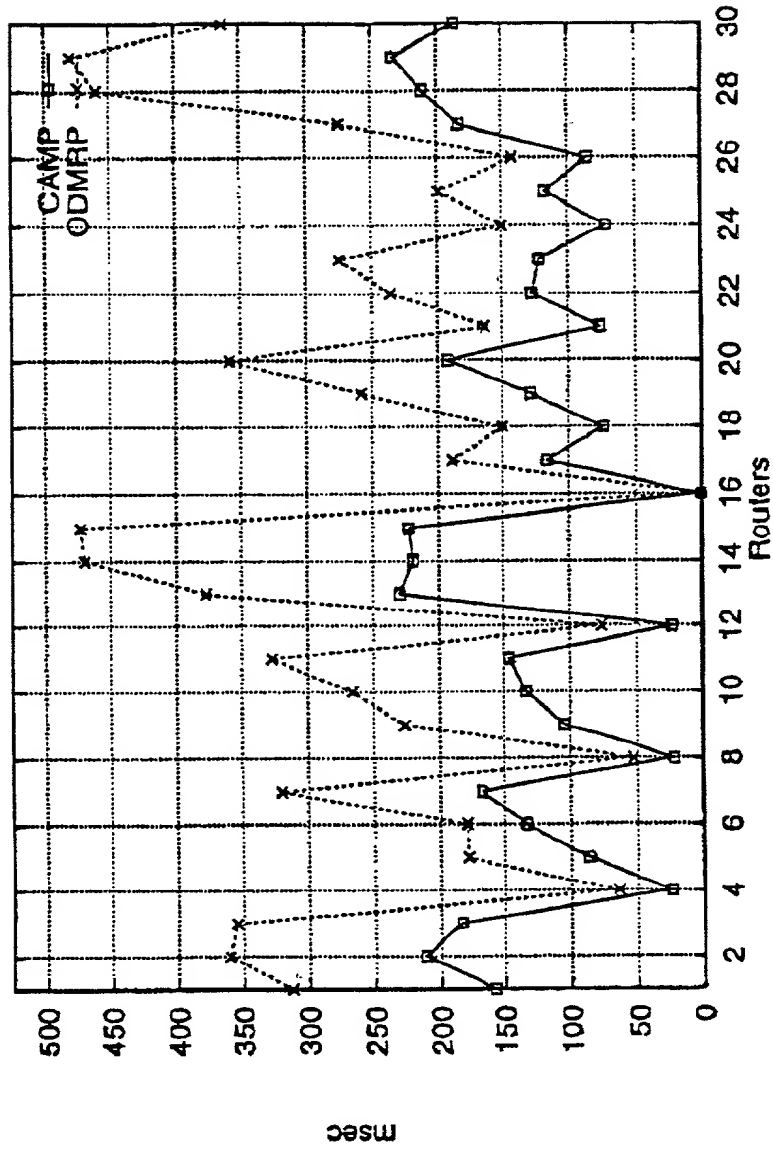


FIG. 12

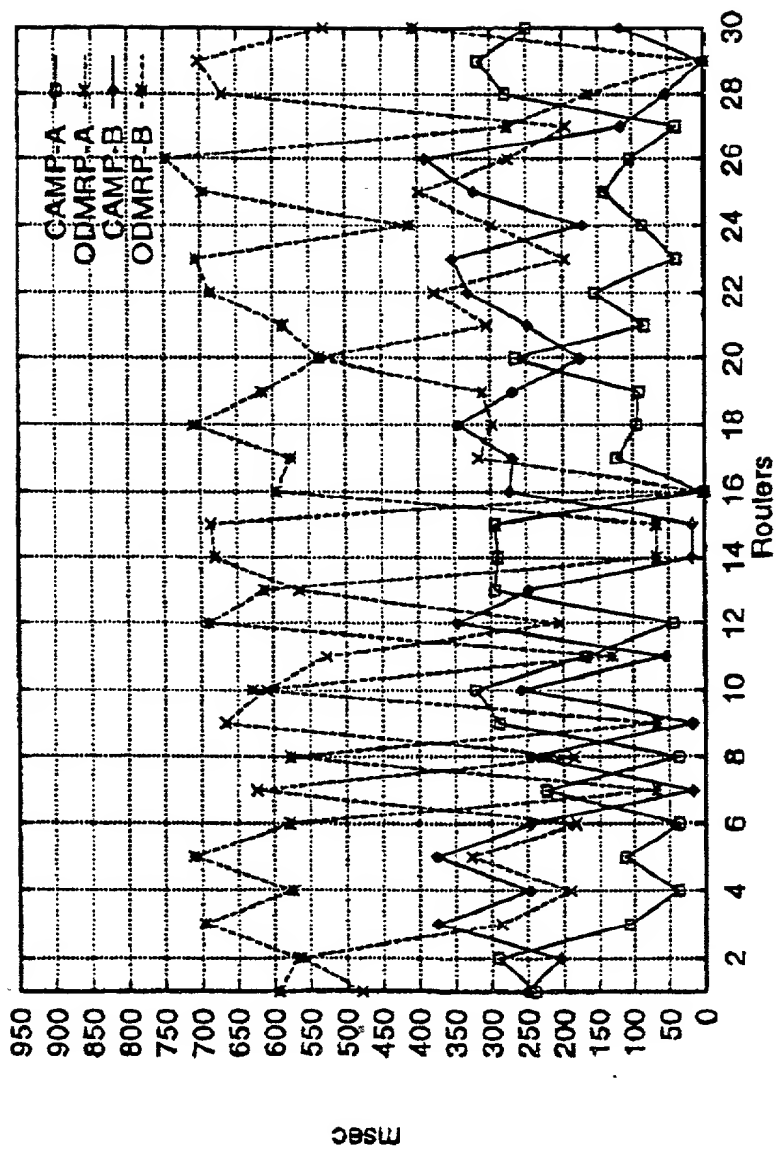


FIG. 13

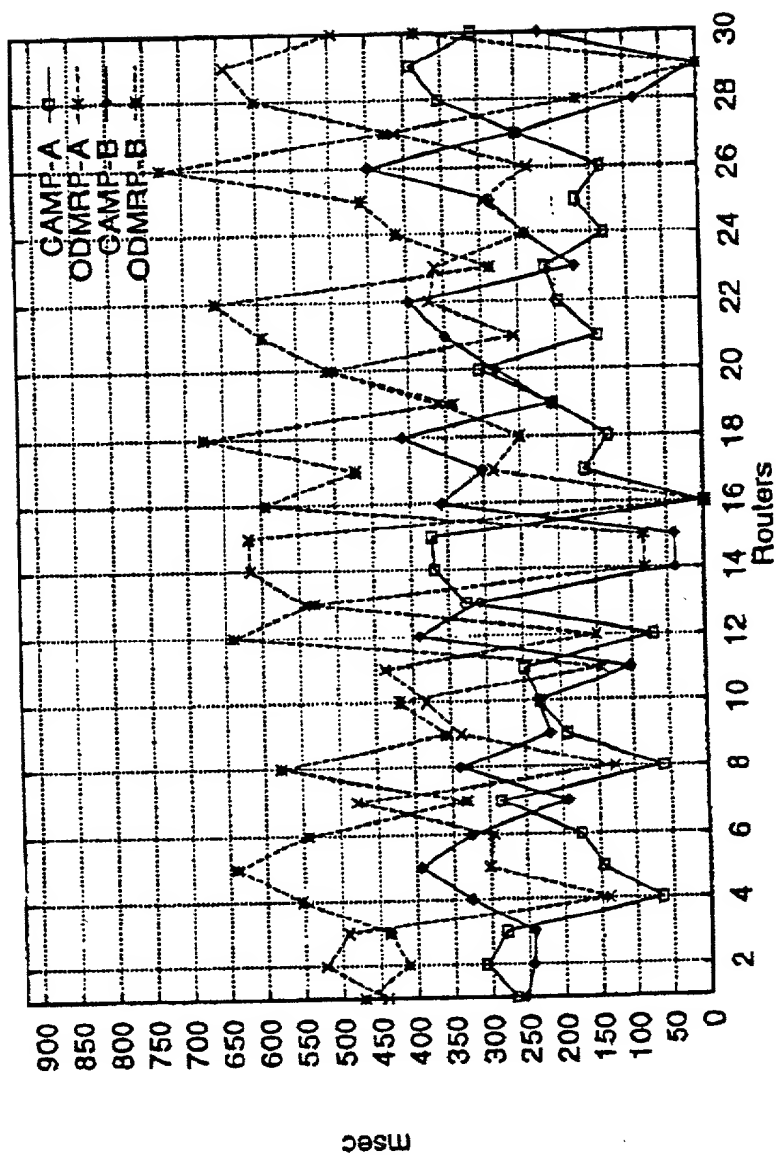


FIG. 14

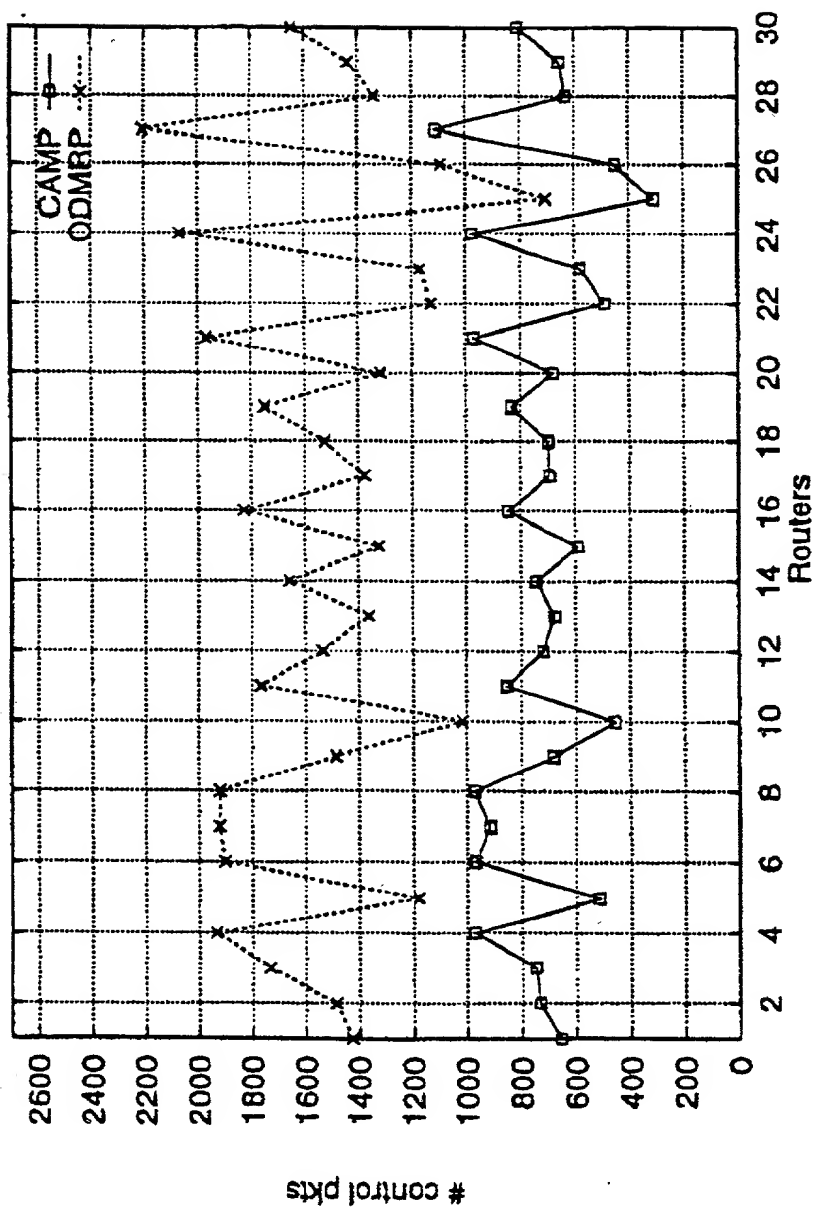


FIG. 15

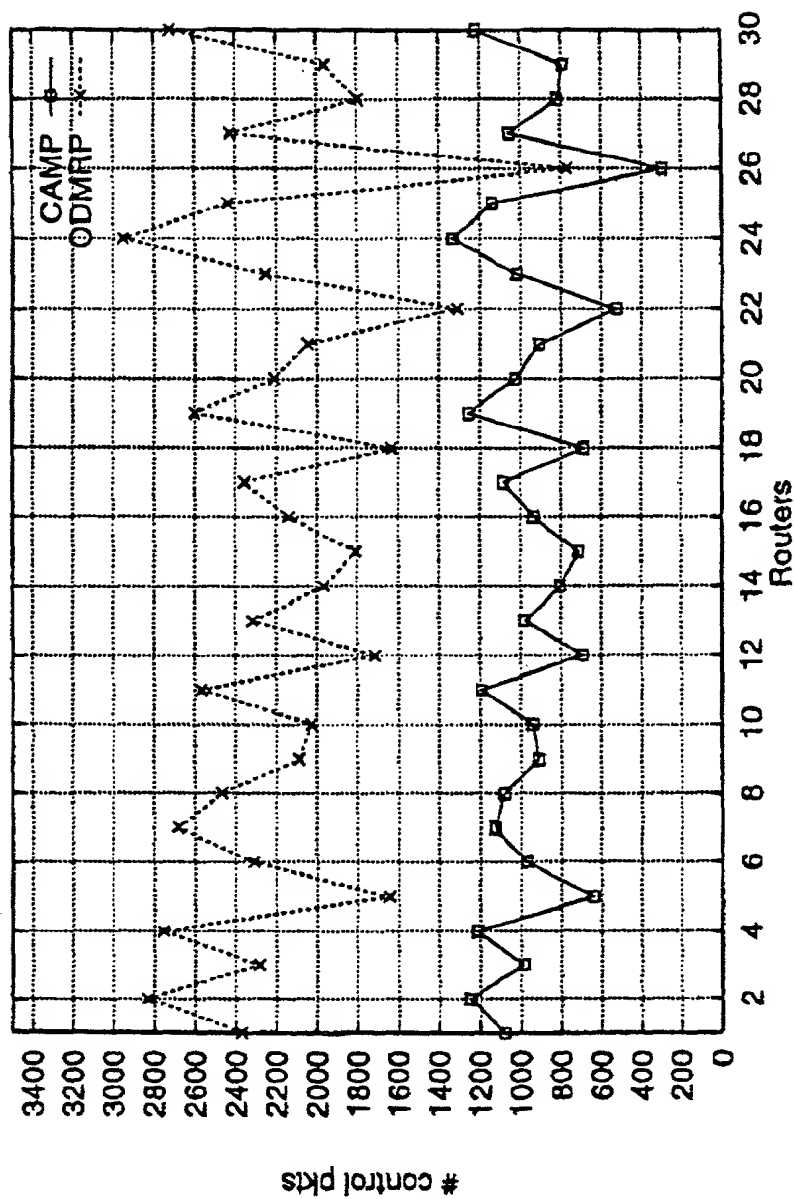


FIG. 16

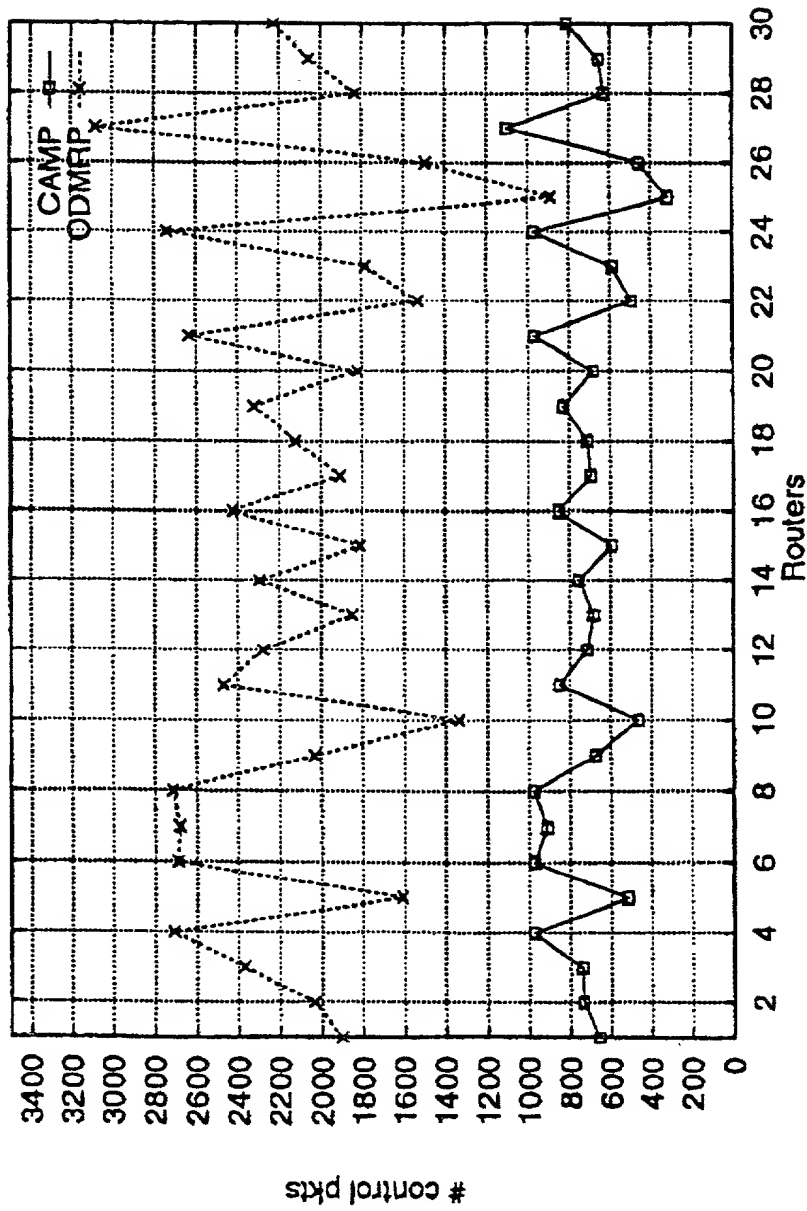


FIG. 17

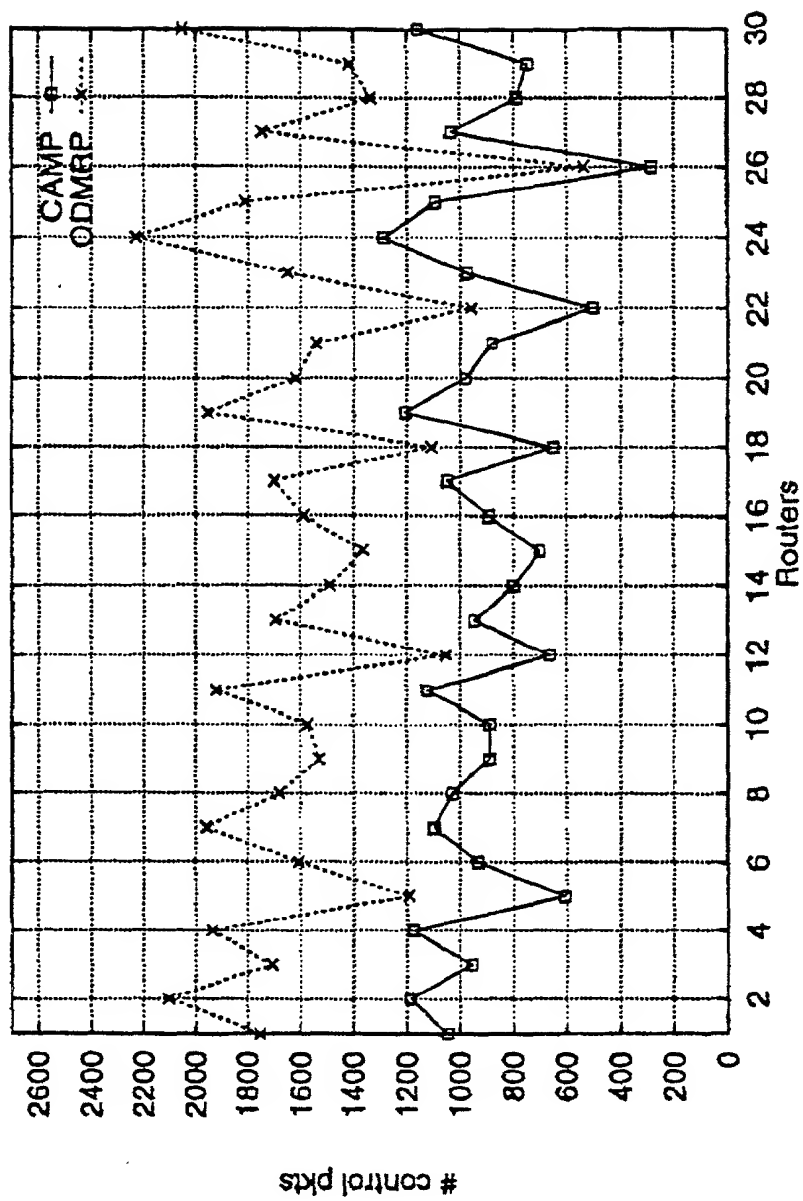


FIG. 18

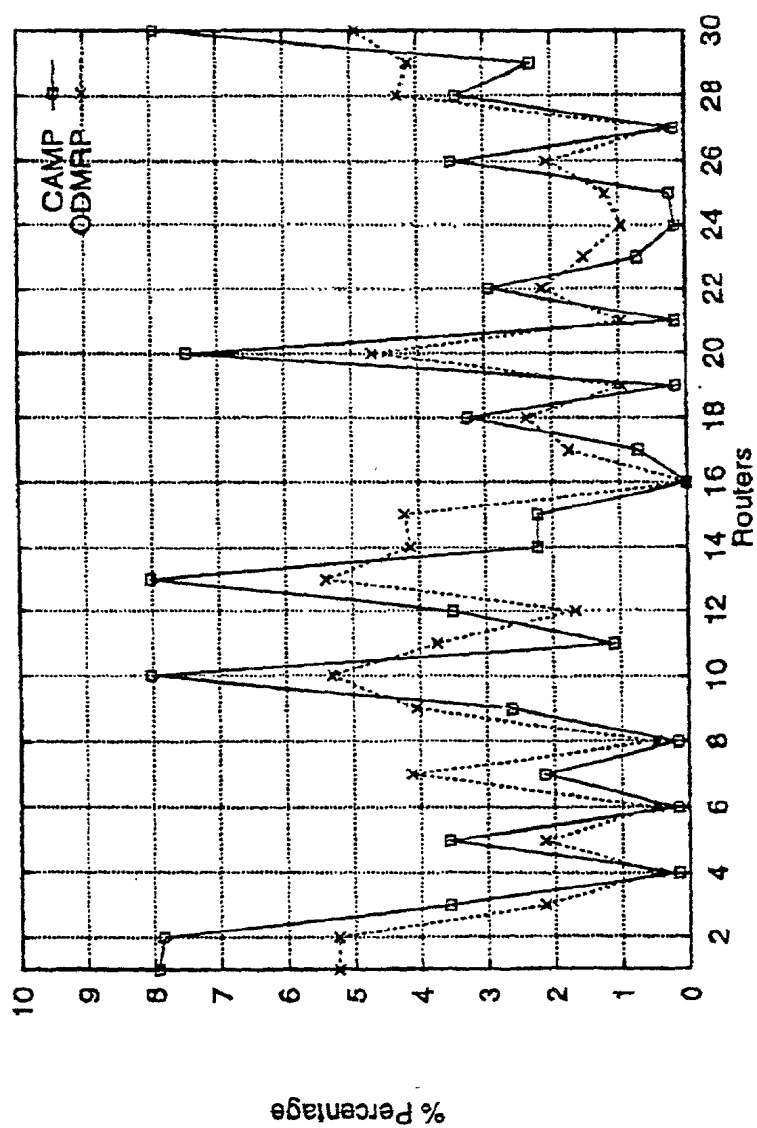


FIG. 19

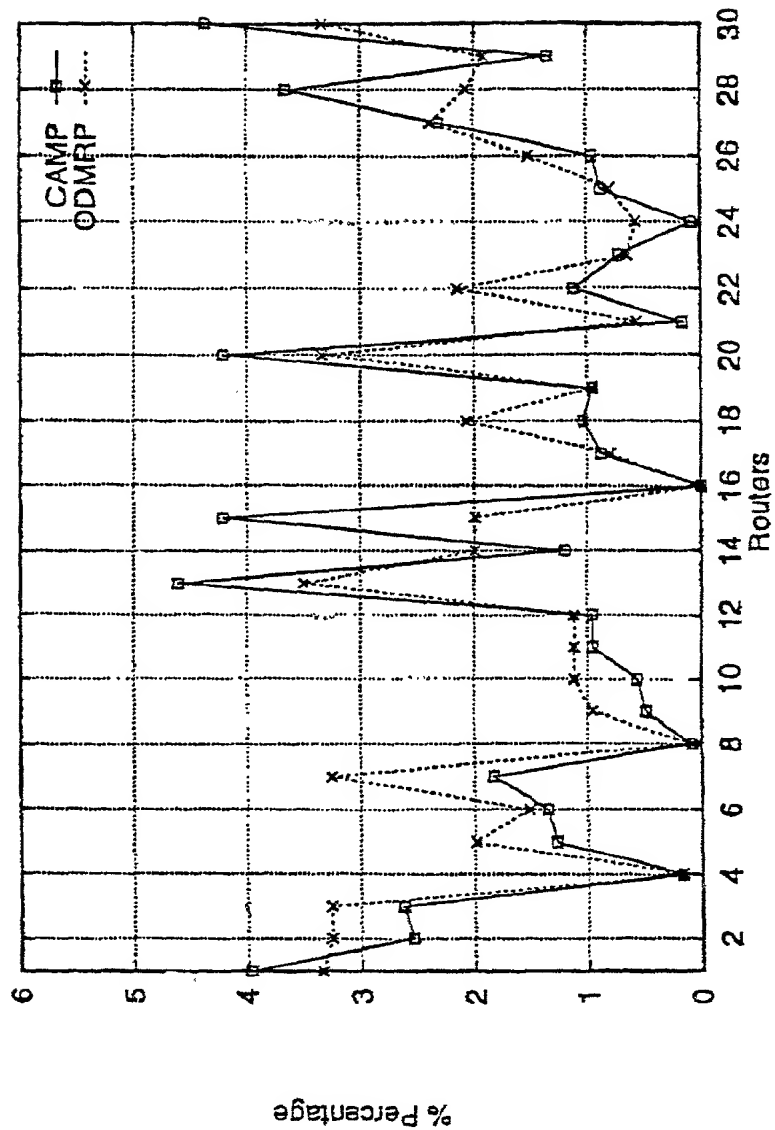


FIG. 20

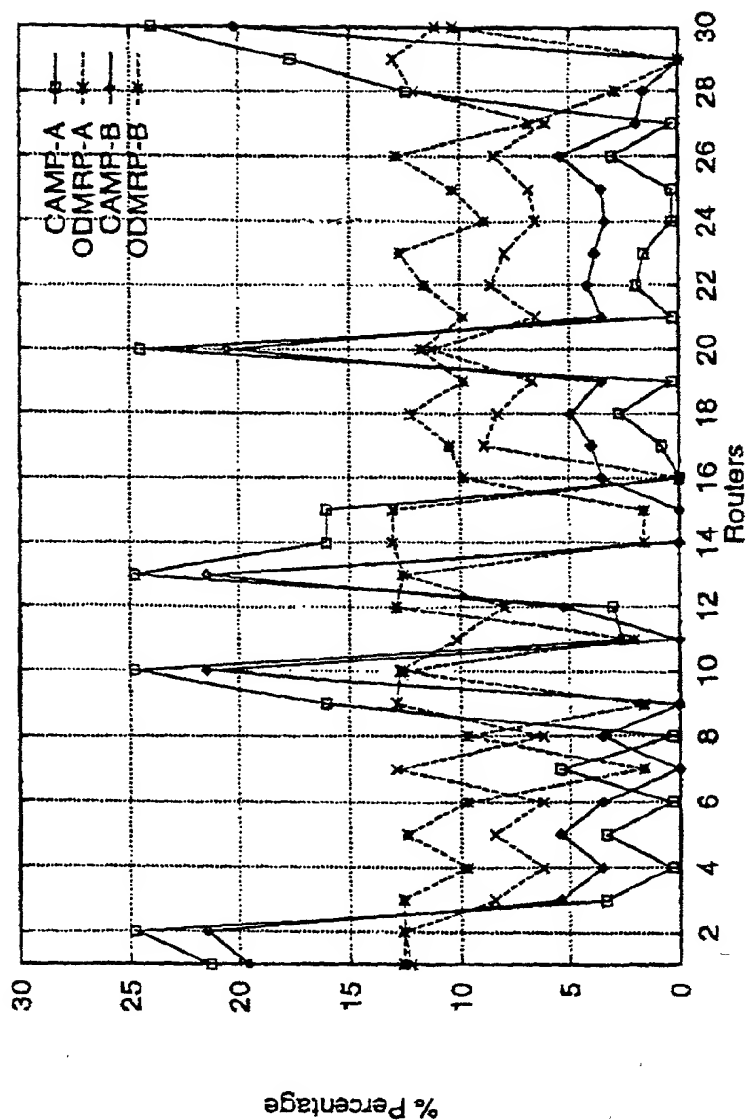


FIG. 21

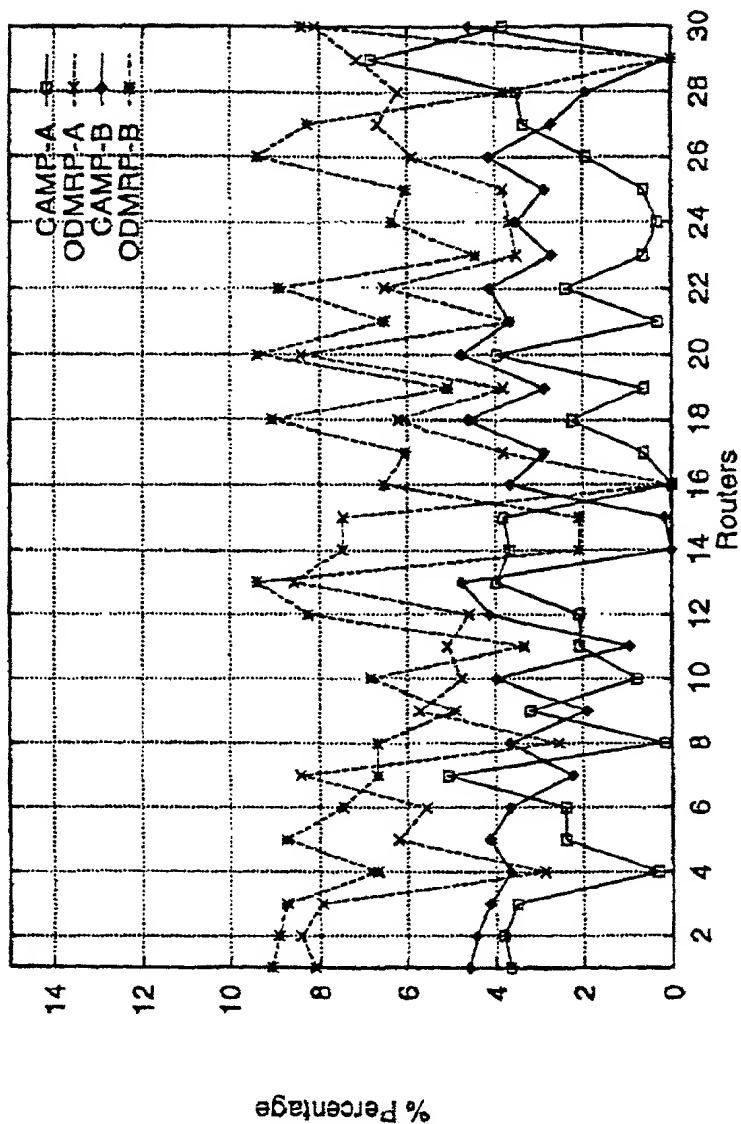


FIG. 22